

第一章：AI-Ready 开发入门

欢迎来到 AI-Ready 开发的世界。在本章中，我们将学习如何构建能够与AI无缝协作的应用程序。

1.1 什么是AI-Ready?

AI-Ready 并不仅仅是调用一个API那么简单，它代表了一种新的开发范式，在这种范式中，应用程序的结构、数据和交互都为AI的介入做好了准备。

随堂练习

- 问题 1: 传统软件开发与 AI-Ready 开发范式最核心的区别是什么？
 - 答案: 最核心的区别在于 **"确定性"**。传统软件的逻辑和产出是高度确定的，而 AI-Ready 应用需要优雅地处理AI模型输出的 **"不确定性"**，并将其转化为可靠的产品特性。
- 问题 2: 为什么在本课程中，我们推荐使用后端生成 Anchor ID 的方案，而不是前端动态生成？
 - 答案: 主要原因是为了保证 **"锚点的稳定性与唯一性"**。
 1. **唯一信源**: 后端是 Markdown 内容的权威来源，由它来解析并生成 ID 可以保证全链路（数据库、API、前端）的一致性。
 2. **避免前端差异**: 不同的前端 Markdown 渲染库甚至不同版本，其 slugify（生成ID）的算法可能存在细微差别，导致锚点链接在某些情况下失效。
 3. **持久化**: 后端生成的 ID 可以被持久化存储，即使未来更换前端技术栈，链接依然有效。
- 问题 3: 在前端将 Marp 嵌套列表转换为 `<details>` 标签有什么好处？
 - 答案: 这样做可以实现 **"内容复用与体验优化"** 的平衡。
 1. **单一内容源**: 我们只需要维护一份对 Marp 工具友好的 Markdown 文稿。
 2. **Web 体验增强**: 在 Web 端，通过 JS 转换，用户可以获得原生、高效的"展开/折叠"交互体验。
 3. **自动化 workflow**: 当需要生成 PPT 时，无需任何修改，Marp 工具就能自动将其识别为带有动画的幻灯片。

1.2 关键原则

构建一个好的AI-Ready应用，需要遵循以下几个原则...